K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are:

1. The centroids of the K clusters, which can be used to label new data

2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically.

Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

K-means algorithm iteratively **minimizes** the distances between every data point and its centroid in order to find the most optimal solution for all the data points.

1. random points of the data set are chosen to be centroids.
2. Distances between every data point and the  centroids are calculated and stored.
3. Based on distance calculates, each point is assigned to the nearest cluster
4. New cluster centroid positions are updated: similar to finding a mean in the point locations
5. If the centroid locations changed, the process repeats from step 2, until the calculated new center stays the same, which signals that the clusters' members and centroids are now set.

Finding the minimal distances between all the points implies that data points have been separated in order to form the most compact clusters possible, with the least variance within them. In other words, no other iteration could have a lower average distance between the centroids and the data points found within them.

**Algorithm 3: *the algorithm of parallel k-means***

***Input****:* the data points *{xi}* and *l* cluster centroids *{x\*}*

***Output****:* members of each cluster

***The map function:***

First, calculate the distances between *{xi}* and *{x\*}*.

Second, find the nearest centroids with each data points. Finally,emit *< x\*, xi >* pairs.

***The reduce function:***

Update *{x\*}* by a mean calculation and obtain the new cluster centroids *{x\*’}*. If *{x\*}* equals to *{x\*’}*, then the clusters are obtained. Otherwise, emit*< x\*, x\* >* pairs as the inputs of the next MapReduce job in accordance with the next iteration.

SCM:

Subtractive clustering assumes that each data point is a potential cluster center. The algorithm does the following:

1. Calculate the likelihood that each data point would define a cluster center, based on the density of surrounding data points.
2. Choose the data point with the highest potential to be the first cluster center.
3. Remove all data points near the first cluster center. The vicinity is determined using [clusterInfluenceRange](https://in.mathworks.com/help/fuzzy/subclust.html?requestedDomain=fr.mathworks.com#inputarg_clusterInfluenceRange).
4. Choose the remaining point with the highest potential as the next cluster center.
5. Repeat steps 3 and 4 until all the data is within the influence range of a cluster center.